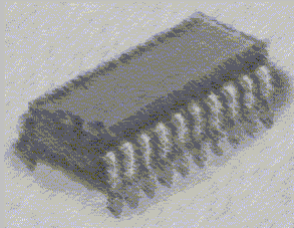


Tema 2. Introducción a los Computadores



Laboratorio de Estructura de Computadores

I. T. Informática de Gestión / Sistemas

Curso 2008-2009

Tema 2:

Transparencia: 2 / 11

Introducción a los computadores

Índice

- Arquitectura Von Neumann
- Fases de ejecución de una instrucción
- Lenguajes de programación
- Simulador MSX88



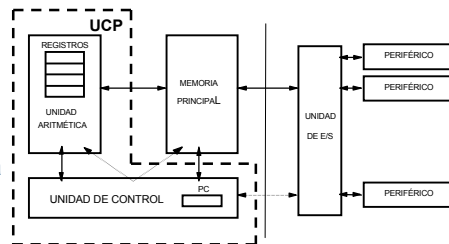
Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Laboratorio de Estructura de Computadores
I. T. I. de Gestión / Sistemas

Arquitectura von Neumann

- Fue establecida en 1945 por John von Neumann
- Su característica principal es que ejecuta instrucciones de máquina de un programa almacenado en memoria

- Bloques:
 - Memoria principal
 - Unidad aritmética y banco de registros
 - Unidad de control (UC)
 - Unidad de entrada/salida



- Los buses son los elementos que interconectan los diferentes elementos de la arquitectura: bus de datos, bus de direcciones y bus de control



Lenguajes de programación (I)

Tipos de lenguajes

- Lenguaje de alto nivel:
 - Posee instrucciones y sintaxis propia (Ej. PASCAL, C)
 - Portabilidad (se compila el mismo código en diferentes máquinas)

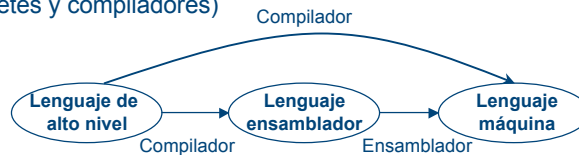
Lenguajes de bajo nivel

- **Lenguaje máquina:**
Las instrucciones de un programa se escriben en binario
 - Incomodo y produce errores ⇔
Solución: usar otros lenguajes de programación
- **Lenguaje ensamblador:**
Las instrucciones se representan con nombres simbólicos o **mnemónicos**
 - Cada instrucción en lenguaje ensamblador se corresponde con una instrucción máquina



Lenguajes de programación (y II)

- Lenguaje de alto nivel (Ejemplo: PASCAL)
BEGIN
Resta:= Minuendo - Sustraendo
END.
- Lenguaje máquina y lenguaje ensamblador (Ejemplo: Ensamblador i80x86)
A10000 MOV AX, Minuendo
2B060200 SUB AX, Sustrayendo
A30400 MOV Resta, AX
- La traducción de un programa a lenguaje máquina lo llevan a cabo los intérpretes y compiladores)



Fases de ejecución de una instrucción

1. **Fase de búsqueda de la instrucción:**
La UC activa las señales de control necesarias para leer de memoria la instrucción a la que apunta el contador de programa (CP)
2. **Fase de decodificación:**
La UC recibe la instrucción (RI) y la decodifica
3. **Búsqueda de operandos:**
La UC, en caso necesario, lee los operandos de memoria o de los registros
4. **Ejecución y almacenamiento del resultado:**
La UC genera las señales necesarias para realizar la operación, y en caso necesario, guarda el resultado en memoria principal o en un registro
5. La UC **actualiza el CP**, para pasar a ejecutar la siguiente instrucción
 - Funcionamiento secuencial
 - Modificación de secuencia ⇔ modificar CP ⇔ bifurcación o salto



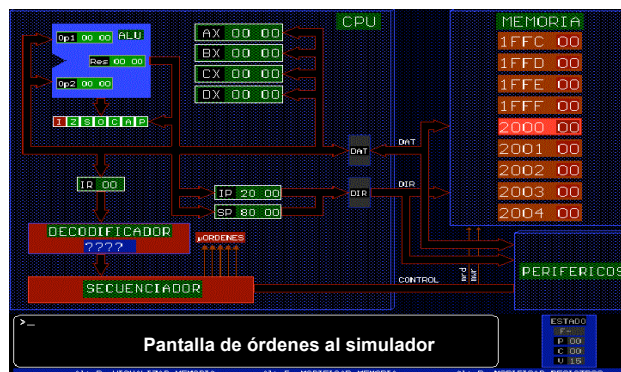
Simulador MSX88 (I)

- El simulador es una versión simplificada del i8088
- Cuenta con una memoria RAM de 64 Kb
- Permite la conexión de periféricos
- Dispone de un programa monitor para el MSX88
- Las instrucciones que ofrece esta CPU son una parte real de las ofrecidas por los microprocesadores i8086 / 8088
- Dispone de las herramientas:
 - ASM88 (ensamblador)
 - LINK88 (montador de programas)



Simulador MSX88 (II)

- La pantalla del simulador es:

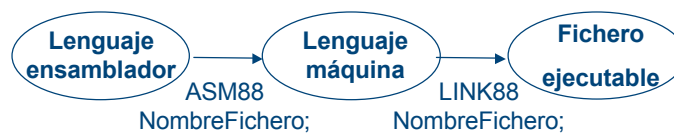


Pantalla de órdenes al simulador



Simulador MSX88 (III)

- Para que se pueda pasar de un fichero en ensamblador al ejecutable se deben seguir los pasos siguientes:
 - Abrir una sesión de MS-DOS
 - Poner el comando: CD C:\MSX88
 - ASM88 NombreFichero;
 - LINK88 NombreFichero;



Simulador MSX88 (y IV)

- Para simular se debe poner:
 - MSX88 (en la pantalla de MS-DOS)
 - L NombreFichero (en la pantalla de órdenes del simulador)
 - R IP 2000h (en la pantalla de órdenes del simulador)
- Con F6 (instrucción a instrucción) o con F7 (ciclo a ciclo de máquina) realizar la simulación
- Q abandona el simulador



Bibliografía

- Fundamentos de los Computadores (Capítulo 1)
Pedro de Miguel Anasagasti
Ed. Paraninfo
- Arquitectura de Computadores (Capítulo 1)
J. Antonio de Frutos, Rafael Rico
Ed. Universidad de Alcalá

